

SERIAL I/O, TIMER, AND INTERFACE CAPABILITIES OF THE MC68901 MULTIFUNCTION PERIPHERAL

Prepared by Geoff Brown System Applications Engineer Motorola, Inc. Austin, Texas

INTRODUCTION

This application note illustrates a system which utilizes several functions of the MC68901 Multifunction Peripheral (MFP). The utilized functions include: 1) USART serial I/O, 2) utilization of internal timers to generate the serial I/O baud rate, 3) utilization of internal timers to generate external interrupts, and 4) use of general purpose I/O pins to provide a cassette interface.

Other general control signal connections are also illustrated. These include: system clock, R/\overline{W} , \overline{IRQ} , \overline{DTACK} , \overline{IACK} , \overline{DS} , \overline{CS} , and timer clock external connections (XTAL1 and XTAL2).

A schematic diagram of the actual hardware used in this application note is shown in Figure 1 at the end of this document. In addition to Figure 1, a listing of the software used with the application is also provided at the end of this document.

HARDWARE CONSIDERATIONS

The hardware shown in Figure 1 uses the MC68008 microprocessor unit (MPU) to control the system; that is, address, data, function codes, data and address strobes, etc. The MC68901 MFP then provides the interrupt and interrupt vectors for the MPU. Eight MCM6665 RAM devices are used to demonstrate the requirement for refresh timing (RAS and CAS). The ROM is implemented in EPROM. Miscellaneous glue parts then tie the system together.

ADDRESS DECODING

Because the addressing range of the MC68008 far exceeds the needs of this application, it is possible to use a simple address decoding scheme. An SN74LS138 3-to-8 demultiplexer (U19) is used to divide the address map into eight 128K segments. Three of these eight segments are assigned to RAM, the MC68901 MFP, and ROM respectively. RAM begins at \$00000, MC68901 MFP begins at \$20000, and ROM (EPROM) begins at \$A0000. The other five segment select control lines are available for expansion.

One problem associated with placing system ROM at any segment other than the bottom of memory is that the MC68008 looks at location \$00000 for its reset vector; however, it is impractical to place ROM at the bottom of the memory map because this would prohibit dynamic interrupt vector programming. This can be resolved by mapping the ROM to the lower portion of memory at reset. In this application, an SN74LS164 shift register (U18) is used to force selection of ROM for the first eight memory cycles after reset to allow the processor to fetch the reset vector and supervisor stack pointer from ROM. When QH of the SN74LS164 shift register is low, selection of ROM is automatic and selection of RAM is inhibited. Once QH goes high, selection proceeds in a normal fashion. U18 is reset whenever HALT and RESET are both active (the system reset condition). Once RESET or HALT become inactive, a logic one is shifted into U18 by the rising edge of \overline{AS} . After eight memory cycles QH goes high and ROM returns to its normal location in the memory map.

RAM CONTROLS

A second SN74LS164 (U17) is used to generate the RAS, CAS, MUX, and DTACK signals. The RAS, CAS, and MUX signals provide control of the dynamic RAM, and DTACK is applied to the MPU to indicate access to the RAM and ROM. Shift register U17 is inhibited from shifting by IACK cycles and by memory cycles to the MC68901. For all other memory cycles, the shift register is allowed to shift and generate DTACK. Notice that DTACK is automatically

generated for all areas of memory other than that assigned to the MC68901 and that only one DTACK time is generated (500 nanoseconds after AS). System performance could be improved by optimizing dynamic RAM sequencing and DTACK generation. RAS is generated for all memory cycles while CAS is enabled by selection of RAM. By generating RAS for all memory cycles it is possible to refresh RAM by executing instructions out of ROM (software refresh). Address multiplexing for the dynamic RAM is accomplished with two SN74LS157 two-input multiplexers (U1 and U2).

MC68008/MC68901 INTERFACE

Interfacing the MC68901 is fairly simple. RESET, DS, R/\overline{W} , and D0-D7 on the MC68901 connect directly to the corresponding pins on the MC68008. RS1-RS5 on the MC68901 connect to the A1-A5 pins on the MC68008. Chip select (\overline{CS}) is generated by qualifying the memory segment signal from U19 with AS. DTACK is gated with the QD output from U17 and passed to the MC68008. The preceeding signals are the only ones that are required for interfacing the MPU with the MFP. In addition, this application utilizes the interrupt capability of the MC68901. The IRO line of the MC68901 is connected directly to both of the MC68008 IPL pins. This corresponds to a level seven interrupt (a nonmaskable interrupt; NMI). Because this application uses the MC68901 to time dynamic refresh intervals, it is imperative that the IRQ interrupt be of the highest priority. If the interrupt capabilities of the MC68901 are to be more fully exploited it is important that no interrupt level be implemented that is higher than the one used for software refresh. The user must never disable or mask the refresh interrupt as this will result in the loss of data. IACK for the MC68901 is generated when the three function codes (FC2-FC0) and A3, A2, and A1 are all high.

For the purpose of baud rate generation, a 2.4576 MHz crystal is connected to the MC68901. Timer C (TCO) is externally connected to the receiver clock (RC) and timer D (TDO) is externally connected to the transmitter clock (TC). Although the software included with this application assumes that the receiver and transmitter clocks operate at the same frequency, the MFP allows for separate clocks.

RESET AND TIMING

The MC68008 requires that an external reset must be applied for at least 100 milliseconds to allow stabilization of the on-chip circuitry and system clock. In this application, system reset is caused at powerup by an MC1455 timer circuit output or it can be generated via a debounced switch. The outputs of the timer and the switch are buffered by open-collector drivers (U27) the outputs of which are connected to $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$.

System timing is provided by a 16 MHz oscillator (U20) which is divided by the two flip flops of U21 to provide 8 MHz (CLK8) and 4 MHz (CLK4) on-chip clocks. The 4 MHz clock is used only by the MC68901 which does not require that its clock be of the same frequency or phase as the system clock.

CASSETTE INTERFACE

Two general purpose I/O lines of the MC68901 (15 and I6) are used for the cassette interface. Data is transmitted and received as square waves. The length of a single cycle of the square wave determines whether a "1" or a "0" is being transferred.

Data for the cassette interface is output at I6 of the MFP.

This output drives a resistor network which divides the voltage by approximately 10. The cassette data output line is then connected to the microphone input of a cassette recorder.

Data to be received from the cassette tape player is shaped in a comparator, U30A. Two IN914 diodes limit the voltage swing to the input of the comparator. The second comparator (U30B) is used to invert the output of U30A. Inversion may or may not be needed depending on whether or not the cassette plays back an inverted signal. The software in this application note assumes that the signal is not inverted. Comparator U30A provides one level of inversion so if the cassette tape player does not provide a level of inversion then a second one must be provided by U30B. The output of comparator U30A is connected to I5 of the MFP (unless U30B is needed).

SOFTWARE

There are six basic software routines included with this application note: MC68901 initialization, software dynamic RAM refresh, transmit character to and receive character from the serial port, transmit character to and receive character from cassette tape. This software represents the basic core of hardware dependent routines necessary for this system.

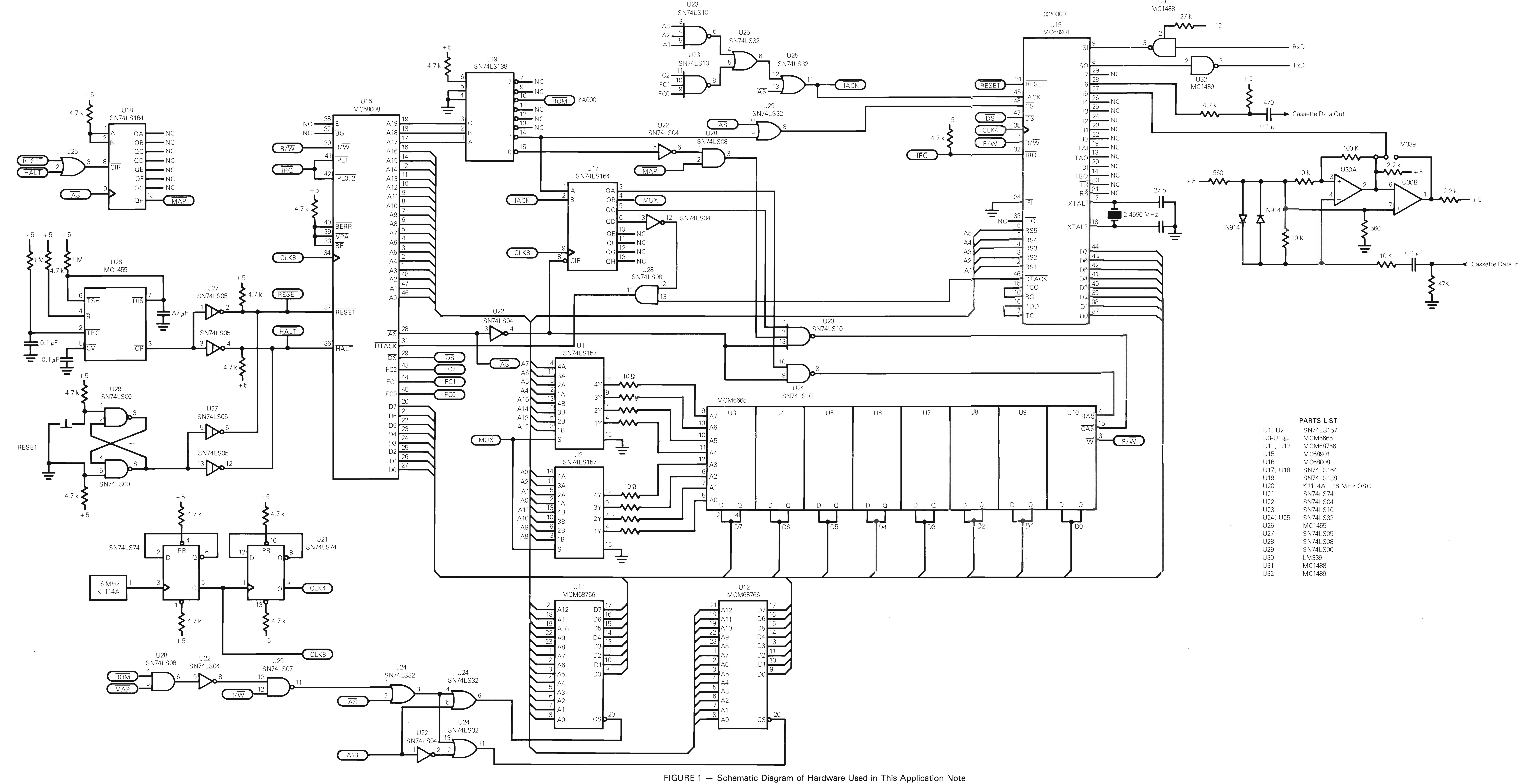
MC68901 INITIALIZATION

Initialization of the MC68901 consists of starting the serial communication clocks, loading the USART control register, and enabling the refresh clock interrupt. Timers C and D are used for serial receiver and transmitter clocks. In this application both timers are programmed for 9600 baud operation. The 2.4576 MHz reference clock is divided by 128 by loading \$02 into both data registers C and D and by starting timers C and D in the divide-by-64 mode. The USART control register is initialized to operate in the divide-by-16 mode (2.4576 MHz/128*16=9600 Hz). In addition, the proper serial communications protocol must be loaded into the USART control register. In this case the USART is programmed for asynchronous communication with: 1 start bit, 1½ stop bits, and odd parity.

In order to facilitate software refresh of dynamic RAM, the MC68901 interrupt vector is initialized and the timer B interrupt enable and mask bits are set. The timer B output serves as the refresh clock.

SOFTWARE REFRESH

Software refresh consists of accessing 128 consecutive memory locations at regularly timed intervals. In this case, it is accomplished by executing 64 NOP instructions of which each requires two memory fetches. The software refresh program is written as a subroutine which may be called at any time to force a refresh. The refresh subroutine resets timer B (the refresh clock) and executes 64 NOP instructions. Timer B is programmed to generate interrupts every 2 milliseconds. The interrupt routine consists simply of a call to the refresh subroutine. One of the main concerns with software refresh is that programs that have critical timing loops (for example the cassette tape interface routines) could be interrupted for refresh if care were not taken. In order to avoid problems, the refresh routine is written so that an interrupt may be forced before a critical timing loop. The user may then be certain that an interrupt will not occur for at least 1.8 milliseconds. A call to the refresh subroutine should be included in any reset routine in order to preclude loss of data.



SERIAL I/O

Both the receive and transmit routines check for break by reading a bit in the receiver status register. If a break is received at any time during serial communications then a jump to a BREAK character handler routine is made. The exact nature of this subroutine is undefined in this application note but it could consist of transmitting a message and then returning to the user's monitor. The transmit routine also checks for a control-W character and halts if one is received. Transmission is then resumed if any character is received. For serial communications, the divide-by-16 mode (a USART control bit) should be used since it results in increased noise rejection. In order to operate the USART in the divide-by-1 mode the receiver clock must be synchronized externally to the received data.

CASSETTE TAPE INTERFACE SOFTWARE

Data is transmitted to the cassette through GPIP6 (bit 6 of the general purpose input/output port control register) and received through GPIP5 (bit 5 of the general purpose input/ output port control register). Data is recorded as a sequence of single cycle square waves with a 500 microsecond period representing a logic one and a one millisecond period representing a logic zero. Before any critical timing loop is executed, in either the transmit or receive routine, a branch to the refresh software is made in order to guarantee that the timing loop will not be interrupted. Timer A of the MC68901 is used for period measurement in both routines. The transmit routine transmits a single byte with the most significant bit first. It is assumed that the first byte of any data stream to be transmitted will be a synchronizing character. In this case the receive routine assumes the synchronizing character to be an ASCII S. The receive routine measures the period length of all incoming square waves in order to generate a bit stream. A simple synchronization routine is included in the program which scans the bit stream for an S. After synchronization data, bytes are assembled from each successive 8-bit block.

MOTOROLA M68000	ASM VERSION	1.30AP	NOTE .S	A 12/14/83	13:30:03	PAGE 1
3 4 5 6 7 8 9 10 11	0.0001.000	* RECE * TRAN * RECE	SMIT CHA IVE CHAR SMIT CHA IVE CHAR SOFTWARE	ACTER FROM RACTER TO T ACTER FROM REFRESH FO	OUGH SERIAL PORT, SERIAL PORT, CAPE, TAPE,	
13 14 15 16 17 18	00001000 00020000 00020001 00020003 00020005 00020007 0002000B	BASE GPIP AER DDR IERA IPRA	ORG EQU EQU EQU EQU EQU EQU	BASE+\$01 BASE+\$03 BASE+\$05 BASE+\$07 BASE+\$0B	MFP BASE ADDRESS GENERAL PURPOSE ACTIVE EDGE DATA DIRECTION INTERRUPT ENABLE INTERRUPT PENDIN	I/O A G A
20 21 22 23 24 25 26	00020013 00020017 00020019 0002001B 0002001D 0002001F 00020021	IMRA VR TACR TBCR TCDCR TADR TBDR	EQU EQU EQU EQU EQU EQU	BASE+\$19 BASE+\$1B BASE+\$1D BASE+\$1F	INTERRUPT MASK A VECTOR TIMER A CONTROL TIMER B CONTROL TIMER C/D CONTRO TIMER A DATA TIMER B DATA	
27 28 29 30 31 32	00020023 00020025 00020029 0002002B 0002002D 0002002F	TCDR TDDR UCR RSR TSR UDR CTLW	EQU EQU EQU EQU EQU EQU	BASE+\$25 BASE+\$29 BASE+\$2B BASE+\$2D	TIMER C DATA TIMER D DATA USART CONTROL RECEIVER STATUS TRANSMITTER STAT USART DATA	us
34 35 36 37 38 39 40		* IN * * * * * *	FOR 960 LOAD US	RANSMITTER O BAUD COMM ART CONTROL		
42 00001000	13FC00020002 0023	INIT	MOVE.B		1/2 TRANSMITTER	
	13FC00020002 0025		MOVE.B		1/2 RECEIVER CL	OCK
	13FC00110002 001D		MOVE.B		DIVIDE BY 4	
	13FC00940002 0029	*	MOVE.B	#\$94,UCR	ODD PARITY, 1 1/	
46 47		*			1 START, ASYNC, 1/16 FOR 9600 B.	I
	13FC00010002 002B		MOVE.B	#\$01,RSR	•	
49 00001028	13FC00050002 002D		MOVE.B	#\$05,TSR	START TRANSMITT	ER CLOCK

MOTORO	LA M68000	ASM VERSION	1.30APN	NOTE .SA	A 12/14/83	13:30:03	PAGE	2
51 52			*	TNTTTAL	IZE REFRESH	INTERRUPT		
53			*	1111111111	tee Kerkeen	111111111111111111111111111111111111111		
54	00001030	13FC00C00002 0017		MOVE.B	#\$C0,VR	LOAD MFP VECTOR	REG	
55	00001038	21FC000010E8 0320		MOVE.L	#RFR2,\$320	LOAD INT VECTOR		
56	00001040	08F900000002 0007		BSET.B	#0,IERA	ENABLE TIMER B I	NT	
57	00001048	08F900000002 0013		BSET.B	#0,IMRA	SET MASK BIT		
58		,	*					
59			*	REFRESH	SUBROUTINE	TO ALLOW SOFTWAR	E	
60			*	TO FORCE	E AN EARLY F	REFRESH		
61			*				•	
62		42390002001B						
63	00001056	13FC00310002 0021		MOVE.B	#49,TBDR	LOAD TIMER B DAT	A REG	
64	0000105E	13FC00060002 001B		MOVE.B	#6,TBCR	START TIMER B 1/	100	
65		00004E71	NOP	EQU	\$4E71			
66	00001066	00004E71	RFI	DCB.W	64,NOP	64 NOPs		
67	000010E6			RTS				
68		6100FF7C	RFR2	BSR RFI		INTERRUPT HANDLE	R	
69 70	000010EC	4E73		RTE		FOR REFRESH		
					**			

MOTORO	LA M68000	ASM VERSION	1.30API	NOTE .S.	A 12/14/83	13:30:03	PAGE	3
72			*					
73 74			*	INPUT C	HARACTER FRO	OM SERIAL PORT I	NTO DO	
75 75	000010EE	083900030002 002B		BTST.B	#3,RSR	(INCH NO ECHO)		
76	00001006	6650	*	D. I. C.		CHECK FOR BREAK		
77 78	000010F6 000010F8	083900070002 002B		BNE.S BTST.B	BREAK #7,RSR	GO PROCESS IT CHECK FOR CHARA	CTER	
79	00001100			BEQ.S	INCHNE	IF NOT READY		
80 81		10390002002F 0200007F		MOVE.B	UDR,D0 #\$7F,D0	READ DATA SIDE DROP PARITY BIT		
82	0000110C			RTS				
83 84			*	SEND CHA	ARACTER IN I	DO.B TO SERIAL P	ORT	
85			*					
86 87	0000110E	6134 083900070002	OUTCH	BSR.S BTST.B	CHKBRK #7,TSR	CHECK FOR BREAK BUFFER EMPTY		
0,		002D		DISTOR	π / / 1510	DOLLEK EMILI		
88	00001118			BEQ.S	OUTCH	STILL NOT READY		
89 90	0000111A	13C00002002F	*	MOVE.B	D0,UDR	SEND CHARACTER		
91			*	CHECK FO	OR CONTROL I	W		
92 93	00001120	083900070002 002B	*	BTST.B	#7,RSR	READ STATUS		
94	00001128	6718		BEQ.S	CTLW9	CHAR NOT READY		
95 96		12390002002F 0C010017		MOVE.B CMP.B	UDR,Dl #CTLW,Dl	READ CHARACTER		
97	00001134	660C		BNE.S	CTLW9	NOT CNTL/W		
98	00001136		CTLWH	BSR.S	CHKBRK	CHECK FOR BREAK		
99	00001138	083900070002 002B		BTST.B	#7,RSR	READ STATUS		
100 101	00001140	67F4	*	BEO	CTLWH	WAIT FOR ANY CH TO CONTINUE	AR	
102 103	00001142	4E75	CTLW9	RTS				
104 105			*	CHECK FO	OR BREAK ON	SERIAL PORT		
106		083900030002 002B	CHKBRK	BTST.B	#3,RSR	READ STATUS		
107 108	0000114C 0000114E			BNE.S RTS	BREAK			
109	00001146	4675	*	KIS				
110			*	WHAT TO	DO WHEN THI	E BREAK IS PRESS	ED	
111 · 112			*					
113	00001150	083900070002 002D	BREAK	BTST.B	#7,TSR	CHECK "TRANSMIT	READY"	
114	00001158			BEQ.S	BREAK	WAIT FOR READY		
115 116		10390002002F 083900030002		MOVE.B BTST.B	UDR,D0 #3,RSR	READ CHARACTER BREAK BUTTON RE	LEASED?	,
117	00001168	002B	*	BNE	BREAK	NO KEEP LOOP		
118 119			*	USER SHO	OULD INSERT	BREAK HANDLER H	ERE	
120 121	0000116A	4E75	*	RTS				

MOTORO	DLA M68000	ASM VERSION	1.30AP	NOTE .S.	A 12/14/83	13:30:03	PAGE	4
123			*					
124 125			*	TRANSMI	T CHARACTER	IN D2 TO TAPE		
126 127 128 129			* * *	PERIOD (OF 1 MILLIS	ORDED AS ONE SOU ECOND DURATION. ONE SQUARE WAVE DURATION.	A LOGI	C.
130 131	0000116C	08F900060002	* TAPE0	BSET.B	#6,DDR	SET GPIP6 AS OU	TPUT	
132		0005 08F900050002			#5,IERA			тет
		0007						, T T
133 134 135 136 137	00001180 00001182 00001186	6100FECC		MOVE.B ROL.B BSR BSR.S	#1,D0 #1,D2 REFRESH TTST	STOP BIT INTO D DATA BIT INTO D FORCE REFRESH WAIT UNTIL PULS HALT TIMER A	2	
138		0019 123C000A			#10,D1	TIMER COUNT FOR	1	
139 140 141	00001194	6606		RNF S	TAPF02	YES		
142	00001196	6606 06810000000A 13C10002001F		ADDI.L	#10,D1	NO. TIMER COUNT		
143 144	0000119C 000011A2	13C10002001F 08F900060002 0001	TAPE02	MOVE.B BSET.B	D1,TADR #6,GPIP	SET TIMER PRELO SEND 1 TO TAPE	AD	
145	000011AA	13FC00050002 0019		MOVE.B -	#\$05,TACR	START TIMER A 1	/64	
146	000011B2	611C		BSR.S	TTST	WAIT UNTIL PULS	E DONE	
147 148		423900020019 08B900060002 0001		CLR.B BCLR.B	TACR #6,GPIP	HALT TIMER SEND 0 TO TAPE		
149	000011C2	13FC00050002 0019		MOVE.B	#\$05,TACR	START TIMER A 1	/64	
150 151 152 153	000011CA 000011CC 000011CE	E300 66B2	*	ASL.B BNE RTS	#1,D0 TAPE01	SENT 8 BITS? NO, CONTINUE		
154 155			*	TIMER T	EST			
156	000011D0	0C3900000002 0019	TTST	CMP.B	#0,TACR	TIMER RUNNING?		
157 158	000011D8 000011DA	083900050002		BEQ.S BTST.B	TTST1 #5,IPRA	NO, RETURN TIME DELAY ELAP	SED?	
159 160	000011E2 000011E4	000B 67EC 08B900050002 000B		BEQ.S BCLR.B	TTST #5,IPRA	NO. WAIT CLEAR INTERRUPT		
161	000011EC		TTSTl	RTS				

MOTORO	LA M68000	ASM VERSION	1.30AP	NOTE .S.	A 12/14/83	13:30:03 PAGE	5
163			*				
164 165 166			* *	RECEIVE	CHARACTER	FROM TAPE INTO DO.B	
167	000011EE	423900020019	TAPEIN	CLR.B	TACR	STOP TIMER A	
168 169	000011F4	4201 083900050002	mi o	CLR.B	DI #5,GPIP	CLEAR DI FOR DATA WAIT FOR LOW	
109	00001110	0001	110	DISI.D	#J,GPIP	WAII FOR LOW	
170	000011FE	66F6 083900050002	m 2 O	BNE	T10	Water Bon Hitou	
171	00001200	0001	120	B121.B	#5,GPIP	WAIT FOR HIGH	
172	00001208	67F6	+	BEQ	T20		
173 174			*	SYNCHRO	NIZE ON S C	HARACTER	
175			*	and the post		TIOD AN AGOTT LOL	
176 177 178			*		TINE LOOKS	FOR AN ASCII 'S' TAPE DATA	
179	0000120A	E301	TS	ASL.B			
180	00001200			BSR.S		GET BIT FROM TAPE	
181 182	0000120E 00001212	0C010053			#'S',D1 TS	S? NO, CONTINUE	
183	00001212				DI,(A6)+	NO, CONTINGE	
184			*				
185			*	GET CHA	RACTER FROM	TAPE	
186 187	00001216	7202		MOVEQ	#2,D1	SET STOP BIT	
188	00001218		GC10	BSR.S		GET BIT FROM TAPE	
189	0000121A	E301		ASL.B	#1,D1	STOP IN CARRY?	
190	0000121C			BCC.S		NO	
191 192	0000121E 00001220			RTS	T30	GET LAST BIT	
193		13FC003B0002	T30		#\$3B,TADR	LOAD TIMER PRELOAD	
194	0000122A	001F 13FC00050002		MOVE B	#5,TACR	START TIMER IN	į
		0019		THE VIDE	* 3 / ALBOIN	CHAINT TIMES IN	
195 196	00001000	(1000010	*	D.CI:D	D 0510 D 044	DIVIDE BY 64 MODE	
196		6100FE1C 083900050002		BSR BTST.B	#5,GPIP	FORCE REFRESH WAIT FOR LOW	
		0001					
198	0000123E		me n	BNE.S	T40	HATE BOD HIGH	
199	00001240	083900050002 0001	TOU	BTST.B	#5,GPIP	WAIT FOR HIGH	
200	00001248			BEO.S	T50		
201 202.	U000124A	423900020019	*	CLR.B	TACR	STOP TIMER	
202.	00001250	16390002001F		MOVE.B	TADR, D3	STORE MEASUREMENT	
204	00001256			CMPI.B	#\$1F,D3	LOGIC 1?	
205	0000125A			BLT.S	T60	MO	
206 207	0000125C		T60	ADDO.B	#1,D1	STORE 1	
207	00001236	7E/J	*	RTS			
209		`	*				
210			*	DME			
211				END			
*****	TOTAL LIN						
****	TOTAL WAR	RNINGS 0					\$

MOTOROLA M68000	0 ASM VERSION 1.30	APNOTE .SA 12/1	14/83 13:30:03	PAGE	6
SYMBOL TABLE L	ISTING				
SYMBOL NAME	SECT VALUE	SYMBOL NAME	SECT VALUE		
AER BASE BREAK CHKBRK CTLW CTLW9 CTLWH DDR GC GC10 GPIP IERA IMRA INCHNE INIT IPRA NOP	00020003 00020000 00001150 00001144 00000017 00001142 00001136 00020005 00001216 00001218 00020001 00020007 00020013 000010EE 00001000 0002000B 00004E71	T20 T30 T40 T50 T60 TACR TADR TAPE0 TAPE01 TAPE02 TAPEIN TBCR TBDR TCDCR TCDR TDDR	00001200 00001222 00001236 00001240 0000125E 0002001F 0000116C 00001180 0000119C 000011EE 0002001B 00020021 00020021 0002001D 00020023 00020025		
OUTCH REFRESH RF1 RFR2 RSR T10	0000110E 00001050 00001066 000010E8 0002002B 000011F6	TSR TTST TTST1 UCR - UDR VR	0002002D 000011D0 000011EC 00020029 0002002F 00020017		

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola and registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Employment Opportunity/ Affirmative Action Employer.



3501 ED BLUESTEIN BLVD., AUSTIN, TEXAS 78721 • A SUBSIDIARY OF MOTOROLA INC.